SgSum:Transforming Multi-document Summarization into Sub-graph Selection

Moye Chen[*], **Wei Li**[*], **Jiachen Liu, Xinyan Xiao, Hua Wu, Haifeng Wang**
Baidu Inc., Beijing, China
{chenmoye,liwei85,liujiachen,xiaoxinyan
wu_hua,wanghaifeng}@baidu.com

Abstract

Most of existing extractive multi-document summarization (MDS) methods score each sentence individually and extract salient sentences one by one to compose a summary, which have two main drawbacks: (1) neglecting both the intra and cross-document relations between sentences; (2) neglecting the coherence and conciseness of the whole summary. In this paper, we propose a novel MDS framework (SgSum) to formulate the MDS task as a *sub-graph selection* problem, in which source documents are regarded as a relation graph of sentences (e.g., similarity graph or discourse graph) and the candidate summaries are its sub-graphs. Instead of selecting salient sentences, SgSum selects a salient sub-graph from the relation graph as the summary. Comparing with traditional methods, our method has two main advantages: (1) the relations between sentences are captured by modeling both the graph structure of the whole document set and the candidate sub-graphs; (2) directly outputs an integrate summary in the form of sub-graph which is more informative and coherent. Extensive experiments on MultiNews and DUC datasets show that our proposed method brings substantial improvements over several strong baselines. Human evaluation results also demonstrate that our model can produce significantly more coherent and informative summaries compared with traditional MDS methods. Moreover, the proposed architecture has strong transfer ability from single to multi-document input, which can reduce the resource bottleneck in MDS tasks.[1]

1 Introduction

Currently, most extractive models treat summarization as a sequence labeling task. They score and select sentences one by one (Zhong et al., 2020). These models (called sentence-level extractors) do not consider summary as a whole but a combination of independent sentences. This may cause incoherent and redundant problem, and result in a poor summary even if the summary consists of high score sentences. Some works (Wan et al., 2015; Zhong et al., 2020) treat summary as a whole unit and try to solve the weakness of sentence-level extractors by using a summary-level extractor. However, these models neglect the intra and cross-document relations between sentences which also have benefits for extracting salient sentences, detecting redundancy and generating overall coherent summaries. Relations become more necessary when input source documents are much longer and more complex such as multi-document input.

In this paper, we propose a novel MDS framework called SgSum which formulates the MDS task as a *sub-graph selection* problem. In our framework, source documents are regarded as a relation graph of sentences (e.g., similarity graph or discourse graph) and the candidate summaries are its sub-graphs. In this view, how to generate a good summary becomes how to select a proper sub-graph. In our framework, the whole graph structure is modeled to help extract salient information from source documents and the sub-graph structures are also modeled to help reflect the quality of candidate summaries. Moreover, the summary is considered as a whole unit, so SgSum directly outputs the final summary in the form of sub-graph. By capturing relations between sentences and evaluating summary as a sub-graph, our framework can generate more informative and coherent summaries compared with traditional extractive MDS methods.

We evaluate SgSum on two MDS datasets with several types of graphs which all significantly improve the MDS performance. Besides, the human evaluation results demonstrate that SgSum can obtain more coherent and informative summaries compared with traditional MDS methods. More-

[*]Equal contribution.

[1]Our code and results are available at: https://github.com/PaddlePaddle/Research/tree/master/NLP/EMNLP2021-SgSum

Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 4063–4074
November 7–11, 2021.

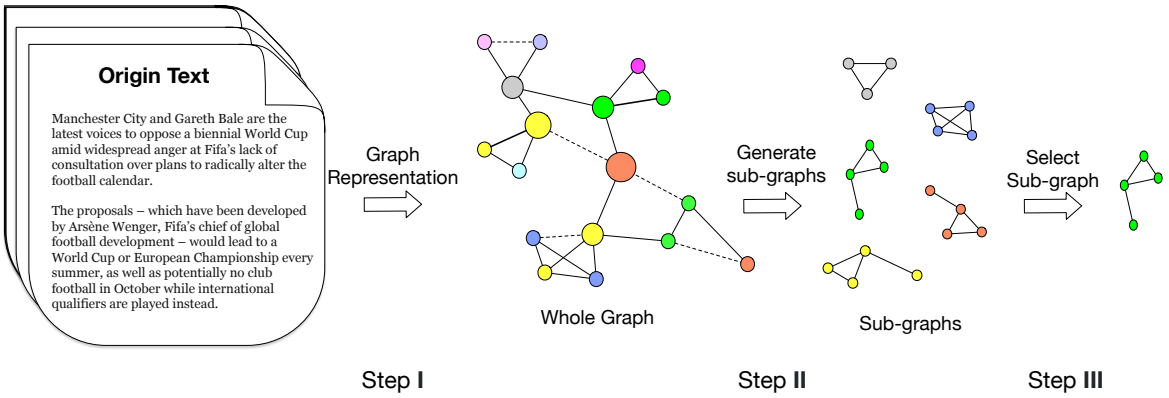

Figure 1: Overview of our sub-graph selection framework. Firstly, well-established graph construction methods are used to transform input documents into a graph where sentences are nodes and semantic links between sentences are edges. Then its sub-graphs can be treated as candidate summaries. Finally, we select the best sub-graph as the final summary.

over, the experimental results also indicate that SgSum has strong power on transfer ability when only trained on single-document data. It performs much better than several strong MDS baselines including supervised and unsupervised models.

The contributions of our work are as follows:

- We propose a novel framework called SgSum which transforms MDS task into the problem of sub-graph selection. The framework leverages graph to capture relations between sentences, and generates more informative and coherent summaries by modeling sub-graph structures.
- Due to the graph-based multi-document encoder, our framework unifies single and multi-document summarization and has strong transfer ability from SDS to MDS task without any parallel MDS training data. Thus, it can reduce the resource bottleneck in MDS tasks.
- Our model is general to several well-known graph representations. We experiment with similarity graph, topic graph and discourse graph on two benchmark MDS datasets. Results show that SgSum has achieved superior performance compared with strong baselines.

2 Summarization as Sub-graph Selection

The graph structure is effective to model relations between sentences which is an essential point to select interrelated summary-worthy sentences in extractive summarization. Erkan and Radev (2004) utilize a similarity graph to construct an unsupervised summarization methods called LexRank. G-Flow (Christensen et al., 2013) and DISCOBERT (Xu et al., 2020) both use discourse graphs to generate concise and informative summaries. Li et al. (2016) and Li and Zhuge (2019) propose to utilize event relation graph to represent documents for MDS. However, most existing graph-based summarization methods only consider the graph structure of source document. They neglect that summary is also a graph and its graph structure can reflect the quality of a summary. For example, in a similarity graph, if selected sentences are lexical similar, the summary is probably redundant. And in a discourse graph, if selected sentences have strong discourse connections, the summary tend to be coherent.

We argue that the graph structure of summary is equally important as the source document. Document graph helps to extract salient sentences, while summary graph helps to evluate the quality of summary. Based on this thought, we propose a novel MDS framework SgSum which transforms summarization into the problem of sub-graph selection. SgSum captures relation of sentences both in whole graph structure (source documents) and sub-graph structures (candidate summaries). Moreover, in our framework, summary is viewed as a whole unit in the form of sub-graph. Thus, SgSum can generate more coherent and informative results than traditional sentence-level extractors.

Figure 1 shows the overview of our framework. Firstly, source documents are transformed into a relation graph by well-known graph construction methods such as similarity graph and discourse graph. Sentences are the basic information units and represented as nodes in the graph. And relations between sentences are represented as edges. For example, a similarity graph can be built based on cosine similarities between tf-idf representations of sentences. Let $\mathbb{G}$ denotes a graph representation matrix of the input documents, where $\mathbb{G}[i][j]$ indicates the tf-idf weights between sentence S_i and S_j.

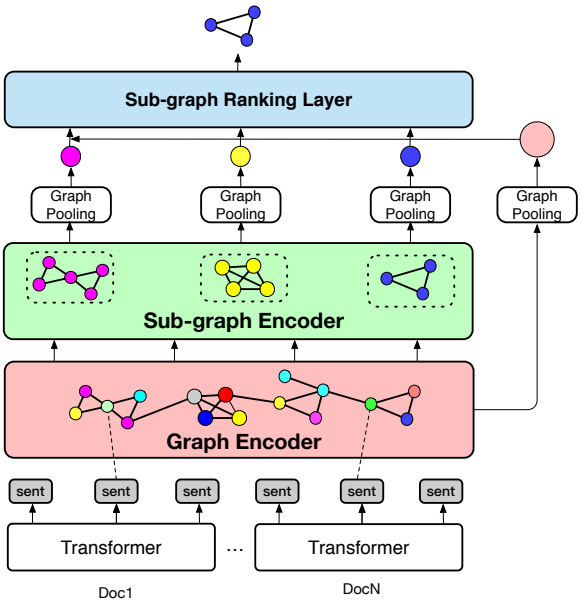

Figure 2: Model architecture of SgSum. Graph-based multi-document encoder takes tokenized documents as input and outputs sentence representations after graph encoding layers. Candidate summaries are modeled by its sub-graph structure in the sub-graph encoder, then scored in a ranking layer.

Formally, the task is to generate the summary S of the document collection given L input sentences $S_1, \ldots, S_L$ and their graph representation $\mathbb{G}$.

As Figure 1 shows, if we represent the source documents as a graph, it can be easily observed that sentences will form plenty of different sub-graphs. By further modelling the sub-graph structures, we can distinguish the quality of different candidate summaries and finally select the best one. Compared with the whole document graph view, sub-graph view is more appropriate to generate a coherent and concise summary. This is also the key point of our framework. Additionally, important sentences usually build up crucial sub-graphs. So it is a simple but efficient way to generate candidate sub-graphs based on those salient sentences.

3 Methodology

3.1 Graph-based Multi-document Encoder

In this section, we introduce our graph-based multi-document encoder. It takes a multi-document set as input and represents all sentences by graph structure. It has three main components: (1) Hierarchical Transformer which processes each document independently and outputs the sentence representations. (2) Graph encoding layer which updates sentence representations by modeling the graph structure of documents. (3) Graph pooling layer which helps to generate an overall representation of source documents. Figure 2 illustrates the overall architecture of SgSum.

Hierarchical Transformer Most previous works (Cao et al., 2017; Jin et al., 2020; Wang et al., 2017) did not consider the multi-document structure. They simply concatenate all documents together and treat the MDS as a special SDS with longer input. Wang et al. (2020) preprocess the multi-document input by truncating lead sentences averagely from each document, then concatenating them together as the MDS input. These preprocessing methods are simple ways to help the model encode multi-document inputs. But they do not make full use of the source document structures. Lead sentences extracted from each document might be similar with each other and result in redundant and incoherent problems. In this paper, we encode source documents by a Hierarchical Transformer, which consists of several shared-weight single Transformers (Vaswani et al., 2017) that process each document independently. Each Transformer takes a tokenized document as input and outputs its sentence representations. This architecture enables our model to process much longer input.

Graph Encoding To effectively capture the relations between sentences in source documents, we incorporate explicit graph representations of documents into the neural encoding process via a graph-informed attention mechanism similar to Li et al. (2020). Each sentence can collect information from other related sentences to capture global information from the whole input. The graph-informed attention mechanism extends the vanilla self-attention mechanism to consider the pairwise relations in explicit graph representations as:

$$\alpha_{ij} = \mathrm{Softmax}(e_{ij} + R_{ij}) \quad (1)$$

where e_{ij} denotes the origin self-attention weights between sentences S_i and S_j, α_{ij} denotes the adjusted weights by graph structure. The key point of the graph-based self-attention is the additional pairwise relation bias R_{ij}, which is computed as a Gaussian bias of the weights of graph representation matrix $\mathbb{G}$:

$$R_{ij} = -\frac{(1 - \mathbb{G}_{ij})^2}{2\sigma^2} \quad (2)$$

where σ denotes the standard deviation that represents the influence intensity of the graph structure. Then a two-layer feed-forward network with ReLU activation and a high-way layer normalization are

applied after the graph-informed attention mechanism. These three components form the graph encoding layers.

Graph Pooling In the MDS task, information is more massive and relations between sentences are much more complex. So it is necessary to have an overview of the central meaning of multi-document input. Zhong et al. (2020) generate a document representation with Siamese-BERT to guide the training and inference process. In this paper, based on the graph representation of documents, we apply a multi-head weighted-pooling operation similar to Liu and Lapata (2019a) to capture the global semantic information of source documents. It takes sentence representations in the source graph as input and outputs an overall representation of them (denoted as D), which provides global information of documents for both the sentence and summary selection processes.

Let x_i denotes the graph representation of sentence S_i. For each head $z \in \{1, ..., n_{head}\}$, we first transform x_i into attention scores a_i^z and value vectors b_i^z, then we calculate an attention distribution $\hat{a}_i^z$ over all sentences in the source graph based on attention scores:

$$a_i^z = W_a^z x_i \tag{3}$$

$$b_i^z = W_b^z x_i \tag{4}$$

$$\hat{a}_i^z = exp(a_i^z) / \sum_{i=1}^{n} exp(a_i^z) \tag{5}$$

We next apply a weighted summation with another linear transformation and layer normalization to obtain vector $head_z$ for the source graph. Finally, we concatenate all heads and apply a linear transformation to ontain the global representation D:

$$head_z = LayerNorm(W_c^z \sum_{i=1}^{n} \hat{a}_i^z b_i^z) \tag{6}$$

$$D = W_d[head_1||...||head_z] \tag{7}$$

where W_a^z, W_b^z, W_c^z and W_d are weight matrices, and $||$ denotes the concatenating operator.

Based on the graph-based multi-document encoder, our model can process much longer input than traditional summarization models. Furthermore, our model can treat SDS and MDS as similar tasks in the unified *sub-graph selection* framework.

3.2 Select from Graph

Sub-graph Encoder As we mentioned in Section 2, sub-graph structure can reflect the quality of candidate summaries. A sub-graph with similar nodes means a redundant summary. And a sub-graph with unconnected nodes represents an incoherent summary. So we apply a sub-graph encoder which has the same architecture with the graph encoder to model each sub-graph. Then we score each sub-graph in a sub-graph ranking layer to select the best sub-graph as the final summary.

Sub-graph Ranking Layer In the training process, we first calculate ROUGE scores of each sentence with the gold summary. Then we select top-K scoring sentences and make a combination of them to form candidate summaries. The sentences in each candidate summary form a subgraph of the source document graph.

There are two principles to optimize our framework. Firstly, a good summary can represent the central meaning of source documents which indicates that a good sub-graph should also represent the whole graph. Specifically, the global document representation D which reflects the overall meaning of source documents should be semantic similar with the gold summary. We use a greedy method (Nallapati et al., 2017) to extract an oracle summary (composed by source sentences) with the largest ROUGE score corresponding to the abstractive reference summary. Then, sentences in the oracle summary are considered as gold summary sentences, which also form a sub-graph. Let C^* denotes the gold summary and the similarity score between C^* and D is measured by $f(D, C^*) = cosine(D, C*)$, which form the following summary-level loss:

$$L_{sum1} = 1 - f(D, C^*) \tag{8}$$

Furthermore, we also design a pairwise margin loss for all the candidate summaries similar with Zhong et al. (2020). We sort all candidate summaries in descending order of ROUGE scores with the gold summary. All candidate summaries are also represented in the form of sub-graph by using sub-graph encoder. Naturally, the candidate pair with a larger ranking gap should have a larger margin, which is the second principle to design our loss function:

$$L_{sum2} = max(0, f(C_j, C^*) - f(C_i, C^*) + \gamma)(i < j) \tag{9}$$

where C_i represents the candidate summary ranked i and γ is a hyperparameter used to distinguish between good and bad candidate summaries. L_{sum1} and L_{sum2} compose a summary-level loss function:

$$L_{sum} = L_{sum1} + L_{sum2} \tag{10}$$

Additionally, we adopt a traditional binary cross-entropy loss between candidate sentences and oracles to learn more accurate sentence and summary representations.

$$L_{sent} = -\sum_{i=1}^{n}(y_i^* \log(\hat{y}_i) + (1 - y_i^*)\log(1 - \hat{y}_i)) \tag{11}$$

where a label $y_i \in \{0, 1\}$ indicates whether the sentence S_i should be a summary sentence. Finally, our loss can be formulated as:

$$L = L_{sent} + L_{sum} \tag{12}$$

During inference, there are hundreds of sentences in a multi-document set which means there are thousands of sub-graphs need to be considered. In order to overcome this difficulty, we adopt a greedy strategy by first selecting several salient sentences as candidate nodes and then making a combination of them to generate candidate sub-graphs. As the important sentences usually build up crucial sub-graphs, it is a simple way to generate candidate sub-graphs based on those salient sentences. Then we calculate cosine similarities between all sub-graphs with the global document representation D in the sub-graph ranking layer, and select the sub-graph with the highest score as the final summary. Thus, our model can be viewed as a *sub-graph selection* framework which means selecting a proper sub-graph from a whole graph.

Furthermore, the graph structure can help to reorder the sentences in the summary to obtain a more coherent summary (Christensen et al., 2013). We order the summary by placing sentences with discourse relations next to each other.

4 Experiments

4.1 Experimental Setup

Graph types We experiment with three well-established graph representations: similarity graph, topic graph and discourse graph. (1) The similarity graph is built based on tf-idf cosine similarities between sentences to capture lexical relations. (2) The topic graph is built based on LDA topic model (Blei et al., 2003) to capture topic relations. The edge weights are cosine similarities between the topic distributions of sentences. (3) The discourse graph is built to capture discourse relations based on discourse markers (e.g. however, moreover), co-reference and entity links as in Christensen et al. (2013). Other types of graphs can also be used in our model. In our experiments, if not explicitly stated, we use the similarity graph by default as it is the most widely used in previous work.

MultiNews Dataset The MultiNews dataset is a large-scale multi-document summarization dataset introduced by (Fabbri et al., 2019). It contains 56,216 articles-summary pairs and each example consists of 2-10 source documents and a human-written summary. Following their experimental settings, we split the dataset into 44,972/5,622/5,622 for training, validation and testing and truncate each document to 768 tokens.

DUC Dataset We use the benchmark datasets from the Document Understanding Conferences (DUC) containing clusters of English news articles and human reference summaries. We use DUC 2002, 2003 and 2004 datesets which contain 60, 30 and 50 clusters of nearly 10 documents respectively. Four human reference summaries have been created for each document cluster by NIST assessors. Our model is trained on DUC 2002, validated on DUC 2003, and tested on DUC 2004. We apply the similar preprocessing method with previous work (Cho et al., 2019) and truncate each document to 768 tokens

Training Configuration We use the base version of RoBERTa (Liu et al., 2019b) to initialize our models in all experiments. The optimizer is Adam (Kingma and Ba, 2014) with $\beta 1$=0.9 and $\beta 2$=0.999, and the learning rate is 0.03 for MultiNews and 0.015 for DUC. We apply learning rate warmup over the first 10000 steps and decay as in (Kingma and Ba, 2014). Gradient clipping with maximum gradient norm 2.0 is also utilized during training. All models are trained on 4 GPUs (Tesla V100) for about 10 epochs. We apply dropout with probability 0.1 before all linear layers. The number of hidden units in our models is set as 256, the feed-forward hidden size is 1,024, and the number of heads is 8. The number of transformer encoding layers and graph encoding layers are set as 6 and 2, respectively. As we mentioned in Section 3.2, during inference we select several salient candidate nodes to build up sub-graphs. And the number of nodes in a sub-graph is determined by the average number of sentences in the gold summary. For MultiNews and DUC, we set the number of candidate nodes and sub-graph nodes as 10/9 and 7/5, respectively.

4.2 Evaluation Results

We evaluate our models on both the MultiNews and DUC datasets to validate their effectiveness on dif-

Models	R-1	R-2	R-L
LexRank	40.27	12.63	37.50
MMR	44.72	14.92	40.07
MatchSum	46.20	16.51	41.89
HeterGraph	46.05	16.35	42.08
PG	43.77	15.38	39.72
Hi-MAP	44.17	16.05	40.35
FT	44.32	15.11	-
GraphSum	46.07	17.42	42.22
SgSum	47.36	18.61	43.13
SgSum(extra)	**47.53**	**18.75**	**43.31**

Table 1: Evaluation results on the MultiNews test set using ROUGE F1[2]. R-1, R-2 and R-L are abbreviations for ROUGE-1, ROUGE-2 and ROUGE-L, respectively.

Models	R-1	R-2	R-L
KLSumm	31.04	6.03	-
LexRank	34.44	7.11	30.95
DPP	38.10	9.14	-
SubModular	38.39	9.58	-
PG	31.43	6.03	-
Sim-DPP	39.35	10.14	-
StructSVM	39.37	9.65	34.52
SgSum	38.66	9.73	34.02
SgSum(extra)	**39.41**	**10.42**	**35.41**

Table 2: Evaluation results on the DUC2004 test set. We report R-1, R-2 and R-L scores, and follow the ROUGE setting of Cho et al. (2019).[3]

ferent types of corpora. The summarization quality is evaluated using ROUGE F1 (Lin, 2004). We report unigram and bigram overlap (ROUGE-1 and ROUGE-2) between system summaries and gold references as a means of assessing informativeness, and the longest common subsequence (ROUGE-L2) as a means of accessing fluency.

Results on MultiNews Table 1 summarizes the evaluation results on the MultiNews. Several strong extractive and abstractive baselines are evaluated and compared with our models. The first block in the table shows results of extractive methods: LexRank (Erkan and Radev, 2004), MMR (Carbonell and Goldstein, 1998), HeterGraph (Wang et al., 2020) and MatchSum (Zhong et al., 2020) which is the previous extractive SOTA model on the MultiNews dataset. The second block shows results of abstractive methods: PG (Lebanoff et al., 2018), Hi-MAP (Fabbri et al., 2019), FT(Flat Transformer) and GraphSum (Li et al., 2020) which is the previous abstractive SOTA model. We report their results following Zhong et al. (2020); Wang et al. (2020); Li et al. (2020). The last block shows the results of SgSum. Compared with both previous extractive and abstractive SOTA models, SgSum achieves more than 1.1/1.2/0.9 improvements on R-1, R-2 and R-L which demonstrates the effectiveness of our *sub-graph selection* framework.

Furthermore, due to our graph representation and graph-based multi-document encoder, our model has the ability to unify single and multi-document summarization task. In our framework, a single document can also be viewed the same as multi-document input. So our model can be enhanced by feeding extra single-document training data. In the last block, *extra* means we leverage CNN/DM data as an extra training resource to improve our model. The results show that single-document data boost the performance of our unified model a further step and achieve a new SOTA result on Multinews.

Results on DUC Table 2 summarizes the evaluation results on the DUC2004 dataset. The first block shows four popular unsupervised baselines, and the second block shows several strong supervised baselines. We report the results of KSLSumm (Haghighi and Vanderwende, 2009), LexRank (Erkan and Radev, 2004), DPP (Kulesza and Taskar, 2011), Sim-DPP (Cho et al., 2019) following Cho et al. (2019). Besides, we also report the results of SubModular (Lin and Bilmes, 2010), StructSVM (Sipos et al., 2012) and PG (See et al., 2017) as strong baselines. The last block shows the results of our models. The results indicate that our model SgSum consistently outperforms most baselines, which further demonstrate the effectiveness of our model on different types of corpora.

Additionally, we also test the performance of SgSum-extra which add CNN/DM data as a supplement. It is comparable to Sim-DPP baseline which also uses extra CNN/DM data to train a similarity model. And the results again show that single-document data greatly improves the performance of our model.

4.3 Transfer Performances

It is commonly known that deep neural networks achieved great improvement on SDS task recently (Liu and Lapata, 2019b; Zhong et al., 2020; Li et al., 2018a,b). However, such supervised models can not work well on MDS task because parallel data for mulit-document are scarce and costly to obtain. For example, the DUC dataset only contains tens of parallel MDS data. There is a pressing need

[2] -n 2 -m -w 1.2 -c 95 -r 1000 -l 250

[3] -n 2 -m -w 1.2 -c 95 -r 1000 -l 100

Models	R-1	R-2	R-L
Lead	40.21	12.13	37.13
LexRank	40.27	12.63	37.50
BERTSUMEXT	41.28	12.05	37.18
SgSum	**43.61**	**14.07**	**39.50**

Table 3: Transfer performance on MultiNews dataset

Models	R-1	R-2	R-L
KLSumm	31.04	6.03	-
LexRank	34.44	7.11	30.95
Extract+Rewrite	28.90	5.33	-
BERTSUMEXT	35.13	8.09	31.28
PG-MMR	36.42	9.36	-
SgSum	**38.18**	**9.46**	**33.81**

Table 4: Transfer performance on DUC2004 dataset

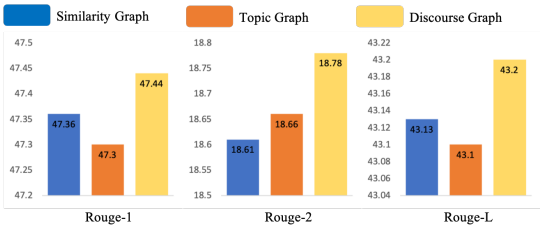

Figure 3: Results on different graph types.

Models	R-1	R-2	R-L
SgSum	**47.36**	**18.61**	**43.13**
w/o s.g. enc	46.87	17.93	42.67
w/o s.g. rank	46.91	17.97	42.80
w/o s.g. enc&rank	46.69	17.64	42.48
w/o graph enc	46.21	17.12	42.11
w/o all	45.43	16.62	41.32

Table 5: Ablation study on the MultiNews test set. s.g. is the abbreviation for sub-graph.

to propose an end-to-end model which is trained on single-document data but can work well with multiple-document input. In this section we do further experiments to verify the transfer ability of our model from single to multi-document task.

We follow the experiment setups of Lebanoff et al. (2018), and compare with several strong baseline models: (1) BERTSUMEXT (Liu and Lapata, 2019b), an extractive method with pre-trained LM model; (2) PG-MMR (Lebanoff et al., 2018), an encoder-decoder model which exploits the maximal marginal relevance method to select representative sentences; (3) Extract+Rewrite (Song et al., 2018), is a recent approach that scores sentences using LexRank and generates a title-like summary for each sentence using an encoder-decoder model. We follow the results of Lebanoff et al. (2018). Table 3 and Table 4 demonstrate the results on MultiNews and DUC2004 respectively.

As shown in Tables 3 and 4, the second blocks are transfer models which are only trained on SDS data and tested on MDS data directly. BERTSUMEXT, PG-MMR, SgSum are trained on CNN/DM, while Extract+Rewrite is trained on Gigaword. The results show that our model achieves better performance than several strong unsupervised models. Furthermore, when trained only on the SDS data, SgSum performs much better on transfer ability compared with the three baselines in the second block of Table 4. The above evaluation results on MultiNews and DUC datasets both validate the effectiveness of our model. The sub-graph selection framework greatly improves the performance of MDS and shows a powerful transfer ability which can reduce the resource bottleneck in MDS.

4.4 Analysis

We further analyze the effects of graph types on our model and validate the effectiveness of different components of our model by ablation studies.

Effects of Graph types We compare the results of similarity graph, topic graph and discourse graph on the MultiNews test set. The comparison results in Figure 3 show that the discourse graph achieves the best performance on all metrics, which demonstrate that graphs with richer relations are more helpful for MDS.

Ablation Study Table 5 summarizes the results of ablation studies, which aim to validate the effectiveness of each individual component of our model. "w/o graph enc" denotes removing the graph-based multi-document encoder, encoding the source input by concatenating all documents as a sequence. "w/o subgraph enc" and "w/o subgraph rank" denontes removing the subgraph encoder and the subgraph ranking layer, respectively. "w/o all" denotes removing all graph components, which is actually the BERTSUMEXT baseline model. The experimental results confirmed that our framework which transforms MDS task into sub-graph selection is effective (see w/o subgraph enc and subgraph rank). Besides, incorporating explicit graph structure (see w/o graph enc) also help to process long input source and result in better performances for MDS.

Models	Informativeness					Coherence				
	1st	2nd	3rd	4th	rating	1st	2nd	3rd	4th	rating
LexRank	0.13	0.14	0.17	0.56	-0.89*	0.09	0.15	0.11	0.65	-1.08*
Submodular	0.29	0.27	0.30	0.14	0.27*	0.31	0.32	0.26	0.11	0.46*
BERTSUMEXT	0.24	0.30	0.29	0.17	0.15*	0.23	0.22	0.41	0.14	-0.01*
SgSum	**0.34**	0.29	0.24	0.13	**0.47**	**0.37**	0.31	0.22	0.10	**0.63**

Table 6: Human evaluation of system summaries on DUC-04. 1st is the best and 4th is the worst. The larger rating denotes better summary quality. * indicates the overall ratings of the corresponding model are significantly (by Welch's t-test with $p < 0.05$) outperformed by our model. The inter-annotator agreement score (Cohen Kappa) is 0.67, which indicates substantial agreement between annotators.

4.5 Human Evaluation

In addition to the automatic evaluation, we also assess system performance by human evaluation. We use the DUC2004 as human evaluation set, and invite 2 annotators to assess the outputs of different models independently. We use Cohen Kappa (Cohen, 1960) to calculate the inter-annotator agreement between annotators. Annotators assess the overall quality of summaries by ranking them considering the following criteria: (1) **Informativeness**: is the main meaning expressed in the source documents preserved in the summary? (2) **Coherence**: is the summary coherent between sentences and well-formed? Annotators were asked to ranking all systems from 1 (best) to 4 (worst). All systems get score 2, 1, -1, -2 for ranking 1, 2, 3, 4 respectively. The rating of each system is computed by averaging the scores on all test instances.

Four system summaries are presented in Table 6. The results demonstrate that SgSum is rated as the best on both informativeness and coherence. Regarding the overall ratings, the summaries generated by SgSum are frequently ranked as the best, which significantly outperforms other models. The human evaluation results further validate the effectiveness of our proposed *sub-graph selection* framework.

5 Related Work

5.1 Graph-based Summarization

Most previous graph extractive MDS approaches aim to extract salient textual units from documents based on graph structure representations of sentences. Erkan and Radev (2004) introduce LexRank to compute sentence importance based on the eigenvector centrality in the connectivity graph of inter-sentence cosine similarity. Christensen et al. (2013) build multi-document graphs to identify pairwise ordering constraints over the sentences by accounting for discourse relationships between sentences. More recently, Yasunaga et al. (2017) build on the approximate discourse graph model and account for macro-level features in sentences to improve sentence salience prediction. Yin et al. (2019) also propose a graph-based neural sentence ordering model, which utilizes an entity linking graph to capture the global dependencies between sentences. Li et al. (2020) incorporate explicit graph representations to the neural architecture based on a novel graph-informed self-attention mechanism. It is the first work to effectively combine graph structures with abstractive MDS model. Wu et al. (2021) present BASS, a novel framework for Boosting Abstractive Summarization based on a unified Semantic graph, which aggregates co-referent phrases distributing across a long range of context and conveys rich relations between phrases. However, these works only consider the graph structure of source documents, but neglect the graph structures of summaries which are also important to generate coherent and informative summaries.

5.2 Sentence or Summary-level Extraction

Extractive summarization methods usually produce a summary by selecting some original sentences in the document set by a sentence-level extractor. Early models employ rule-based methods to score and select sentenecs (Lin and Hovy, 2002; Lin and Bilmes, 2011; Takamura and Okumura, 2009; Schilder and Kondadadi, 2008). Recently, SUMMARUNNER (Nallapati et al., 2017) adopt an encoder based on Recurrent Neural Networks which is the earliest neural summarization model. SUMO (Liu et al., 2019a) capitalizes on the notion of structured attention to induce a multi-root dependency tree representation of the document. However, all these models belong to sentence-level extractors which select high score sentences individually and might raise redundancy (Narayan et al., 2018).

Different from above studies, some work focus on the summary-level selection. Wan et al. (2015) optimize the summarization performance directly

based on the characteristics of summaries and rank summaries directly during inference. Bae et al. (2019), Paulus et al. (2017) and Celikyilmaz et al. (2018) use reinforcement learning to globally optimize summary-level performance. Recent studies (Alyguliyev, 2009; Galanis and Androutsopoulos, 2010; Zhang et al., 2019) have attempted to a build two-stage document summarization. The first stage is usually to extract some fragments of the original text, and the second stage is to select or modify on the basis of these fragments. Mendes et al. (2019) follow the extract-then-compress paradigm to train an extractor for content selection. Zhong et al. (2020) propose a novel extract-then-match framework which employs a sentence extractor to prune unnecessary information, then outputs a summary by matching models. These methods consider summary as a whole rather than individual sentences. However, they neglect the relations between sentences during both scoring and selecting.

5.3 From Single to Multi-document

Recent neural network summarization models focus on SDS due to the large parallel datasets automatically harvested from online news websites including Gigaword (Rush et al., 2017), CNN/DM (Hermann et al., 2015), NYT (Sandhaus, 2018) and Newsroom (Grusky et al., 2018). However, MDS has not yet fully benefited from the development of neural network models, because parallel data for MDS are scarce and costly to obtain.

A promising route to generating summary from a multi-document input is to apply a model trained for SDS to a "mega-document" (Lebanoff et al., 2018) created by concatenating all documents together. Nonetheless, such a model may not suit well for two reasons. First, identifying important text pieces from a mega-document can be challenging for the model, which is trained on single-document data where the summary-worthy content is often contained in the first few sentences. This is not the case for a mega-document. Second, redundant text pieces in a mega-document can be repeatedly used for summary generation under the current framework. Lebanoff et al. (2018) present a novel adaptation model, named PG-MMR, to generate summary from multi-document inputs. However, it still considers MDS data as a meta-document. In contrast, our model unifies SDS and MDS by graph representations, and achieves great performance on transferring from SDS to MDS.

Conclusion

We propose a novel framework SgSum which transforms the MDS task into the problem of sub-graph selection. SgSum captures the relations between sentences by modelling both the graph structure of the whole document set and the candidate sub-graphs, then directly output an integrate summary in the form of sub-graph which is more informative and coherent. Experimental results on two MDS datasets show that SgSum brings substantial improvements over several strong baselines. Moreover, the proposed architecture has strong transfer ability from single to multi-document, which can reduce the resource bottleneck in MDS tasks.

Acknowledgments

This work was supported in part by the National Key R&D Program of China under Grant 2020YFB1406701.

References

RM Alyguliyev. 2009. The two-stage unsupervised approach to multidocument summarization. *Automatic Control and Computer Sciences*, 43(5):276–284.

Sanghwan Bae, Taeuk Kim, Jihoon Kim, and Sanggoo Lee. 2019. Summary level training of sentence rewriting for abstractive summarization. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, pages 10–20, Hong Kong, China. Association for Computational Linguistics.

David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022.

Ziqiang Cao, Wenjie Li, Sujian Li, and Furu Wei. 2017. Improving multi-document summarization via text classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31.

Jaime Carbonell and Jade Goldstein. 1998. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 335–336.

Asli Celikyilmaz, Antoine Bosselut, Xiaodong He, and Yejin Choi. 2018. Deep communicating agents for abstractive summarization. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1662–1675, New Orleans, Louisiana. Association for Computational Linguistics.

Sangwoo Cho, Logan Lebanoff, Hassan Foroosh, and Fei Liu. 2019. Improving the similarity measure of determinantal point processes for extractive multi-document summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1027–1038, Florence, Italy. Association for Computational Linguistics.

Janara Christensen, Stephen Soderland, Oren Etzioni, et al. 2013. Towards coherent multi-document summarization. In *Proceedings of the 2013 conference of the North American chapter of the association for computational linguistics: Human language technologies*, pages 1163–1173.

Jacob Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement, 20, 37–46.*

Günes Erkan and Dragomir R Radev. 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of artificial intelligence research*, 22:457–479.

Alexander Fabbri, Irene Li, Tianwei She, Suyi Li, and Dragomir Radev. 2019. Multi-news: A large-scale multi-document summarization dataset and abstractive hierarchical model. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1074–1084, Florence, Italy. Association for Computational Linguistics.

Dimitrios Galanis and Ion Androutsopoulos. 2010. An extractive supervised two-stage method for sentence compression. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 885–893.

Max Grusky, Mor Naaman, and Yoav Artzi. 2018. Newsroom: A dataset of 1.3 million summaries with diverse extractive strategies. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 708–719, New Orleans, Louisiana. Association for Computational Linguistics.

Aria Haghighi and Lucy Vanderwende. 2009. Exploring content models for multi-document summarization. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 362–370, Boulder, Colorado. Association for Computational Linguistics.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. *Advances in neural information processing systems*, 28:1693–1701.

Hanqi Jin, Tianming Wang, and Xiaojun Wan. 2020. Multi-granularity interaction network for extractive and abstractive multi-document summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6244–6254.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980.*

Alex Kulesza and Ben Taskar. 2011. Learning determinantal point processes.

Logan Lebanoff, Kaiqiang Song, and Fei Liu. 2018. Adapting the neural encoder-decoder framework from single to multi-document summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4131–4141, Brussels, Belgium. Association for Computational Linguistics.

Wei Li, Lei He, and Hai Zhuge. 2016. Abstractive news summarization based on event semantic link network. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 236–246. The COLING 2016 Organizing Committee.

Wei Li, Xinyan Xiao, Jiachen Liu, Hua Wu, Haifeng Wang, and Junping Du. 2020. Leveraging graph to improve abstractive multi-document summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6232–6243. Association for Computational Linguistics.

Wei Li, Xinyan Xiao, Yajuan Lyu, and Yuanzhuo Wang. 2018a. Improving neural abstractive document summarization with explicit information selection modeling. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pages 1787–1796.

Wei Li, Xinyan Xiao, Yajuan Lyu, and Yuanzhuo Wang. 2018b. Improving neural abstractive document summarization with structural regularization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4078–4087.

Wei Li and Hai Zhuge. 2019. Abstractive multi-document summarization based on semantic link network. *IEEE Transactions on Knowledge and Data Engineering*.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Chin-Yew Lin and Eduard Hovy. 2002. From single to multi-document summarization. In *Proceedings of the 40th annual meeting of the association for computational linguistics*, pages 457–464.

Hui Lin and Jeff Bilmes. 2010. Multi-document summarization via budgeted maximization of submodular functions. In *Human Language Technologies:*

The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, pages 912–920, Los Angeles, California. Association for Computational Linguistics.

Hui Lin and Jeff Bilmes. 2011. A class of submodular functions for document summarization. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 510–520.

Yang Liu and Mirella Lapata. 2019a. Hierarchical transformers for multi-document summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5070–5081, Florence, Italy. Association for Computational Linguistics.

Yang Liu and Mirella Lapata. 2019b. Text summarization with pretrained encoders. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3730–3740, Hong Kong, China. Association for Computational Linguistics.

Yang Liu, Ivan Titov, and Mirella Lapata. 2019a. Single document summarization as tree induction. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1745–1755.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019b. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Afonso Mendes, Shashi Narayan, Sebastião Miranda, Zita Marinho, André F. T. Martins, and Shay B. Cohen. 2019. Jointly extracting and compressing documents with summary state representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3955–3966, Minneapolis, Minnesota. Association for Computational Linguistics.

Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. 2017. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Ranking sentences for extractive summarization with reinforcement learning. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1747–1759, New Orleans, Louisiana. Association for Computational Linguistics.

Romain Paulus, Caiming Xiong, and Richard Socher. 2017. A deep reinforced model for abstractive summarization. *arXiv preprint arXiv:1705.04304*.

Alexander M Rush, SEAS Harvard, Sumit Chopra, and Jason Weston. 2017. A neural attention model for sentence summarization. In *ACLWeb. Proceedings of the 2015 conference on empirical methods in natural language processing*.

Evan Sandhaus. 2018. The new york times annotated corpus. *Linguistic Data Consortium, Philadelphia, 6(12)*.

Frank Schilder and Ravikumar Kondadadi. 2008. FastSum: Fast and accurate query-based multi-document summarization. In *Proceedings of ACL-08: HLT, Short Papers*, pages 205–208, Columbus, Ohio. Association for Computational Linguistics.

Abigail See, Peter Liu, and Christopher Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Association for Computational Linguistics*.

Ruben Sipos, Pannaga Shivaswamy, and Thorsten Joachims. 2012. Large-margin learning of submodular summarization models. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 224–233, Avignon, France. Association for Computational Linguistics.

Kaiqiang Song, Lin Zhao, and Fei Liu. 2018. Structure-infused copy mechanisms for abstractive summarization. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1717–1729.

Hiroya Takamura and Manabu Okumura. 2009. Text summarization model based on the budgeted median problem. In *Proceedings of the 18th ACM conference on Information and knowledge management*, pages 1589–1592.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Xiaojun Wan, Ziqiang Cao, Furu Wei, Sujian Li, and Ming Zhou. 2015. Multi-document summarization via discriminative summary reranking. *arXiv preprint arXiv:1507.02062*.

Danqing Wang, Pengfei Liu, Yining Zheng, Xipeng Qiu, and Xuanjing Huang. 2020. Heterogeneous graph neural networks for extractive document summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6209–6219, Online. Association for Computational Linguistics.

Kexiang Wang, Tianyu Liu, Zhifang Sui, and Baobao Chang. 2017. Affinity-preserving random walk for multi-document summarization. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 210–220, Copenhagen, Denmark. Association for Computational Linguistics.

Wenhao Wu, Wei Li, Xinyan Xiao, Jiachen Liu, Ziqiang Cao, Sujian Li, Hua Wu, and Haifeng Wang. 2021. BASS: Boosting abstractive summarization with unified semantic graph. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6052–6067, Online. Association for Computational Linguistics.

Jiacheng Xu, Zhe Gan, Yu Cheng, and Jingjing Liu. 2020. Discourse-aware neural extractive text summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5021–5031, Online. Association for Computational Linguistics.

Michihiro Yasunaga, Rui Zhang, Kshitijh Meelu, Ayush Pareek, Krishnan Srinivasan, and Dragomir Radev. 2017. Graph-based neural multi-document summarization. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 452–462, Vancouver, Canada. Association for Computational Linguistics.

Yongjing Yin, Linfeng Song, Jinsong Su, Jiali Zeng, Chulun Zhou, and Jiebo Luo. 2019. Graph-based neural sentence ordering. *arXiv preprint arXiv:1912.07225*.

Haoyu Zhang, Jingjing Cai, Jianjun Xu, and Ji Wang. 2019. Pretraining-based natural language generation for text summarization. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 789–797, Hong Kong, China. Association for Computational Linguistics.

Ming Zhong, Pengfei Liu, Yiran Chen, Danqing Wang, Xipeng Qiu, and Xuanjing Huang. 2020. Extractive summarization as text matching. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6197–6208, Online. Association for Computational Linguistics.